

## Mesh projection between parametric surfaces

X. Roca, J. Sarrate<sup>\*,†</sup> and A. Huerta

*Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III,  
Universitat Politècnica de Catalunya, Jordi Girona 1-3, E-08034 Barcelona, Spain*

### SUMMARY

This paper presents a new algorithm to map a given mesh over a source surface onto a target surface. This projection is determined by means of a least-squares approximation of a transformation defined between the loops of boundary nodes of the cap surfaces in the parametric spaces. Once the new mesh is obtained on the parametric space of the target surface, it is mapped to the target surface according to its parameterization. Therefore, in contrast with the usual techniques, the developed algorithm does not require solving any root finding problem to ensure that the projected nodes are on the target surface. Finally, this projection algorithm is extended to three-dimensional cases and included in a sweep meshing tool in order to generate the inner layers of elements in the physical space.

KEY WORDS: finite element method; mesh generation; hexahedral elements; sweep

### 1. INTRODUCTION

Hexahedral mesh generation techniques have been improved in the last decade. Several algorithms have been devised in order to generate hexahedral meshes for any arbitrary geometry (see References [1,2] for a detailed survey). However, a general and fully automatic hexahedral mesh generation algorithm is still an unreachable goal. Moreover, further research is still needed in order to work out a general purpose algorithm that, given any volume, generates high quality hexahedral elements at low cost (both in cpu and in user interaction time). Therefore, special attention has been focused on existing algorithms that decompose the entire geometry into several simpler pieces (assembly models). These smaller volumes can

---

\*Correspondence to: J. Sarrate, Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III, Universitat Politècnica de Catalunya, Jordi Girona 1-3, E-08034 Barcelona, Spain.

†E-mail: jose.sarrate@upc.edu

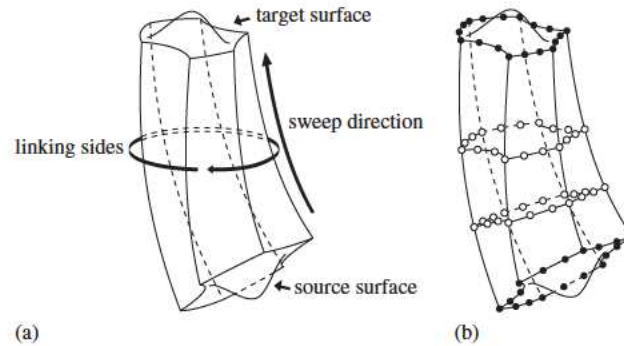


Figure 1. (a) Generic one-to-one sweepable volume; and (b) available data for generating the inner layers of nodes: • boundary nodes of the cap surfaces, and ○ boundary nodes of the inner layers.

be easily meshed by well-known methods that exhibit an outstanding performance in these simpler volumes [3–7].

Most of the commercial CAD packages allow to model volumes by extruding, or sweeping, a surface along a delimited axis. These one-to-one sweep volumes are defined by a *source surface*, a *target surface* and a series of *linking-sides* (see Figure 1(a)). In order to ensure that a given geometry is one-to-one sweepable, the following conditions must be satisfied:

- (i) The source and target surfaces must be topologically equivalent (they must have the same number of holes and logical sides). However, they may have different areas and curvatures.
- (ii) The linking-sides must be mappable, or equivalently, defined by four logical sides.
- (iii) The sweep volume must be defined by only one axis.
- (iv) The sweep volume must be defined by only one axis.

A detailed presentation on constraints which must be met for a volume to be sweepable, in a generic sense, are presented in Reference [8]. Based on the definition of an extrusion geometry, the traditional procedure to generate an all-hexahedral mesh by sweeping consists of the following four steps:

- (i) Generation of a quadrilateral mesh over the source surface (structured or not).
- (ii) Projection of the source mesh onto the target surface.
- (iii) Generation of a structured quadrilateral mesh over the linking-sides.
- (iv) Generation of the inner layers of nodes and elements.

Several quadrilateral surface mesh generation algorithms can be used in the first step [9–12]. The gridding of the linking-sides involved in the third step can be generated using any standard structured quadrilateral surface mesh generator [2, 13]. Hence, the two main issues to be dealt with by any sweep algorithm are the second and fourth step. In both steps, the meshes to be generated (i.e. the mesh over the target surface and the inner layer meshes) must be topologically equivalent to the source surface mesh.

The standard procedure to map meshes between surfaces involves orthogonal projections of nodes onto the target surface [4, 14]. These projections are expensive from a computational

point of view since it is necessary to solve as many root finding problems as internal points there are on the grid of the source surface. In order to overcome this drawback, we present a new and efficient algorithm to map a given mesh over the source surface onto the target surface. This projection is determined by means of a least-squares approximation of an affine mapping defined between the parametric representation of the loops of boundary nodes of the cap surfaces. Once the new mesh is obtained on the parametric space of the target surface, it is mapped to the target surface according to its parameterization.

The developed algorithm to map meshes between cap surfaces cannot be directly applied in order to generate the inner layer of nodes, since these layers are not defined by parametric surfaces. In fact, the available data to determine the position of the inner layers of nodes is: (i) the loops of nodes on the linking surfaces and (ii) the cap surface meshes (see Figure 1(b)). Hence, the previous projection algorithm is extended to the three-dimensional space and it is used to generate the inner layers of elements in the physical space. The resulting algorithm becomes analogous to the method proposed in Reference [3]. Hence, the inner nodes are located using a weighted least-squares approximation of the transformation between the boundary nodes of the cap surfaces and the boundary nodes of the layer as in Reference [5].

## 2. CAP SURFACES MESH GENERATION

Commercial CAD packages are usually applied to define the geometry in industrial applications. Since cap surfaces may be parameterized trimmed surfaces, the quadrilateral mesh generator used to discretize them has to be able to work with trimmed surfaces. In our application, the source surface is discretized using an extended version to parametric surfaces of our unstructured quadrilateral mesh generator [11, 12].

Once the source surface,  $S$ , is meshed, the next step in the sweep algorithm is to map it onto the target surface,  $T$ . This paper presents a new and efficient algorithm to map meshes between trimmed surfaces. In fact, the mapping is defined between the parametric domain of the surfaces,  $D_S$  and  $D_T$ . Then, the obtained mesh is mapped to the target surface according to its parameterization. For some surfaces, it may be necessary to smooth the new surface mesh. Note that this smoothing is also needed in other methods [14].

First of all, we will show that determining a projection between trimmed surfaces is equivalent to looking for a projection between their parametric spaces. To this end, assume that the source and target surfaces are trimmed surfaces. Let

$$\boldsymbol{\varphi}_S : V_S \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad \boldsymbol{\varphi}_T : V_T \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

be their extended parameterization, where  $V_S$  and  $V_T$  are two open and bounded sets of  $\mathbb{R}^2$ . Note that the domain of a trimmed surface is, in general, not a rectangle  $[a, b] \times [c, d] \subset \mathbb{R}^2$  in the parametric space. If  $\boldsymbol{\varphi}_S$  and  $\boldsymbol{\varphi}_T$  are continuous and injective, the Brouwer's theorem on invariance of domain [15] states that they are also open mappings. Since they are open mappings, their restrictions (that is, the definition of the trimmed surfaces)

$$\boldsymbol{\varphi}_S|_{D_S} : D_S \subset V_S \rightarrow S \subset \mathbb{R}^3, \quad \boldsymbol{\varphi}_T|_{D_T} : D_T \subset V_T \rightarrow T \subset \mathbb{R}^3$$

are homeomorphisms in  $D_S$  and  $D_T$ , respectively. Hence, we have

$$S = \boldsymbol{\varphi}_S(D_S), \quad T = \boldsymbol{\varphi}_T(D_T)$$



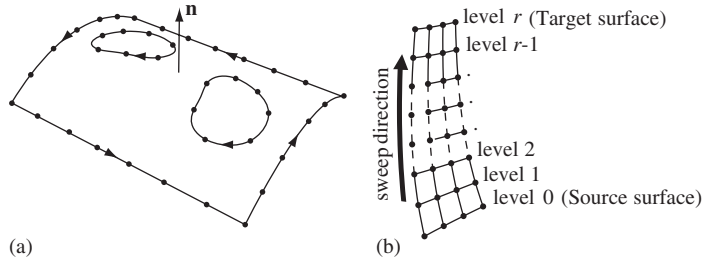


Figure 2. (a) Loops of nodes on the boundary of a non-simple connected surface; and (b) linking sides are discretized using  $r-1$  inner levels.

Recall that our aim is to determine a mapping  $\tilde{\psi}: S \rightarrow T$  such that, given a mesh,  $\mathcal{M}_S$ , over the source surface, it yields a mesh,  $\mathcal{M}_T$ , onto the target surface with the same connectivities. Since  $S$  and  $T$  have the same topology we can assume that  $\tilde{\psi}$  is also a homeomorphism.

Taking into account that  $\phi_S|_{D_S}$ ,  $\phi_T|_{D_T}$  and  $\tilde{\psi}$  are homeomorphisms, it is possible to define

$$\psi := \phi_T|_{D_T}^{-1} \circ \tilde{\psi} \circ \phi_S|_{D_S}$$

such that

$$\begin{array}{ccc} S \subset \mathbb{R}^3 & \xrightarrow{\tilde{\psi}} & T \subset \mathbb{R}^3 \\ \phi_S|_{D_S} \uparrow & & \uparrow \phi_T|_{D_T} \\ D_S \subset \mathbb{R}^2 & \xrightarrow{\psi} & D_T \subset \mathbb{R}^2 \end{array} \quad (1)$$

Under these conditions, the diagram of mappings (1) is a commutative diagram. Hence, it is feasible to find first the projection  $\psi$ , homeomorphism between the parametric domains  $D_S$  and  $D_T$ , and then, mapping the new mesh onto the target surface,  $T$ , according to its parameterization,  $\phi_T|_{D_T}$ , as

$$\tilde{\psi} = \phi_T|_{D_T} \circ \psi \circ \phi_S|_{D_S}^{-1} \quad (2)$$

Note that we do not need the analytical expression of the inverse function  $\phi_S|_{D_S}^{-1}$ . It suffices to store the pre-images of nodal co-ordinates of  $\mathcal{M}_S$  by  $\phi_S|_{D_S}$ . This can be achieved if the application stores both the physical and the parametric co-ordinates of each mesh node.

Therefore, special attention is focused on the definition of the projection between  $D_S$  and  $D_T$  from the available data. Let  $m$ , with  $m \geq 3$ , be the number of nodes on all the boundary loops of the cap surfaces. We assume that each cap surface is delimited by one outer boundary and one inner boundary for each hole. These boundaries are previously meshed, and a series of loops of nodes on the boundary of each surface is obtained (see Figure 2(a)). Let  $U_S = \{\mathbf{u}_S^i\}_{i=1,\dots,m} \subset \mathbb{R}^2$  and  $U_T = \{\mathbf{u}_T^i\}_{i=1,\dots,m} \subset \mathbb{R}^2$  be the parametric co-ordinates of all boundary nodes of the source and target surfaces, respectively. It is important to point out that the physical co-ordinates of these points (i.e. their images by  $\phi_S|_{D_S}$  and  $\phi_T|_{D_T}$ ) do not necessarily determine planar loops. The goal is to find a function  $\psi$  such that

$$\psi(\mathbf{u}_S^i) = \mathbf{u}_T^i, \quad i = 1, \dots, m \quad (3)$$

In this new algorithm, the homeomorphism  $\psi$  is approximated by an affine mapping

$$\mathbf{u}_T = \psi(\mathbf{u}_S) \approx \mathbf{A}(\mathbf{u}_S - \mathbf{u}_S^{\text{arb}}) + \mathbf{b} \quad (4)$$

where  $\mathbf{u}_S$  and  $\mathbf{u}_T$  are points on  $D_S$  and  $D_T$ , respectively,  $\mathbf{A}$  is a linear transformation with the origin at one arbitrary point  $\mathbf{u}_S^{\text{arb}}$  and  $\mathbf{b}$  is a translation vector. Unfortunately, given any two loops of boundary data there is not, in general, an affine mapping that verifies (3). Therefore, we look for a linear transformation,  $\mathbf{A}$ , and a translation vector  $\mathbf{b}$  that fit conditions (3) in the least-squares sense. Hence,  $\mathbf{A}$  and  $\mathbf{b}$  are such that minimize

$$F(\mathbf{A}, \mathbf{b}) = \sum_{i=1}^m \|\mathbf{u}_T^i - (\mathbf{A}(\mathbf{u}_S^i - \mathbf{u}_S^{\text{arb}}) + \mathbf{b})\|^2 \quad (5)$$

It is straightforward to show that if

$$\mathbf{u}_S^{\text{arb}} := \mathbf{u}_S^c = \frac{1}{m} \sum_{i=1}^m \mathbf{u}_S^i$$

then

$$\mathbf{b} = \mathbf{u}_T^c = \frac{1}{m} \sum_{i=1}^m \mathbf{u}_T^i \quad (6)$$

Therefore, the following co-ordinates are defined

$$\bar{\mathbf{u}}_S = \mathbf{u}_S - \mathbf{u}_S^c, \quad \bar{\mathbf{u}}_T = \mathbf{u}_T - \mathbf{u}_T^c \quad (7)$$

such that (4) can be written as

$$\bar{\mathbf{u}}_T = \bar{\psi}(\bar{\mathbf{u}}_S) \approx \mathbf{A} \bar{\mathbf{u}}_S \quad (8)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

where  $\bar{\psi}$  is the expression of  $\psi$  in the new co-ordinates. Now the minimization problem (5) is

$$F(\mathbf{A}) = \sum_{i=1}^m \|\bar{\mathbf{u}}_T^i - \mathbf{A} \bar{\mathbf{u}}_S^i\|^2 \quad (9)$$

Since  $\mathbf{A}$  is a linear transformation, it can be written as

$$\mathbf{A} = \sum_{i=1}^4 \lambda_i \mathbf{h}_i \quad (10)$$

where, given  $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ ,

$$\mathbf{h}_1(\mathbf{u}) := \begin{pmatrix} u_1 \\ 0 \end{pmatrix}, \quad \mathbf{h}_2(\mathbf{u}) := \begin{pmatrix} u_2 \\ 0 \end{pmatrix}, \quad \mathbf{h}_3(\mathbf{u}) := \begin{pmatrix} 0 \\ u_1 \end{pmatrix}, \quad \mathbf{h}_4(\mathbf{u}) := \begin{pmatrix} 0 \\ u_2 \end{pmatrix} \quad (11)$$

is a basis of the linear transformations from  $\mathbb{R}^2$  to  $\mathbb{R}^2$ , and  $\lambda_i \in \mathbb{R}$ , with  $i = 1, \dots, 4$ .

If the scalar product of two functions

$$\mathbf{f} : U_S \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\mathbf{g} : U_S \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

is defined as

$$\langle \mathbf{f}, \mathbf{g} \rangle := \sum_{i=1}^m \langle \mathbf{f}(\mathbf{u}_S^i), \mathbf{g}(\mathbf{u}_S^i) \rangle_{\mathbb{R}^2} = \sum_{i=1}^m \mathbf{f}(\mathbf{u}_S^i)^T \cdot \mathbf{g}(\mathbf{u}_S^i) \quad (12)$$

then the normal equations of the least-squares problem (9) are

$$\mathbf{N}\boldsymbol{\lambda} = \mathbf{d} \quad (13)$$

where  $N_{i,j} = \langle \mathbf{h}_i, \mathbf{h}_j \rangle = \sum_{k=1}^m \mathbf{h}_i(\mathbf{u}_S^k)^T \cdot \mathbf{h}_j(\mathbf{u}_S^k)$  and  $d_i = \langle \mathbf{h}_i, \tilde{\boldsymbol{\psi}} \rangle = \sum_{k=1}^m \mathbf{h}_i(\mathbf{u}_S^k)^T \cdot \mathbf{u}_T^k$ , with  $i = 1, \dots, 4$  and  $j = 1, \dots, 4$ .

The matrix  $\mathbf{N}$  is non-singular if and only if not all the points in  $U_S$  are aligned. In practical applications, the source points are not aligned because they lie on the boundary of a non-degenerated surface. Therefore, system (13) has a unique solution,  $\boldsymbol{\lambda}^0$ . From the numerical solution,  $\boldsymbol{\lambda}^0$ , of the linear system (13) the linear transformation  $\mathbf{A}^0 = \sum_{i=1}^4 \lambda_i^0 \mathbf{h}_i$  can be determined. Therefore, using  $\mathbf{A}^0$  and Equations (4) and (6), the following affine mapping can be established:

$$\boldsymbol{\psi}^0(\mathbf{u}_S) := \mathbf{A}^0(\mathbf{u}_S - \mathbf{u}_S^c) + \mathbf{u}_T^c \quad (14)$$

In conclusion, an affine mapping (14) between parametric spaces has been found that fits, in the least-squares sense, the loops of boundary data. This transformation can be used to map meshes from  $D_S$  to  $D_T$ . Finally, to obtain the mesh  $\mathcal{M}_T$  we only need to map the nodes to the target surface  $T$ . To this end, and according to (2), we define

$$\tilde{\boldsymbol{\psi}}^0(\mathbf{p}) := \boldsymbol{\varphi}_T|_{D_T}(\boldsymbol{\psi}^0(\boldsymbol{\varphi}_S|_{D_S}^{-1}(\mathbf{p}))), \quad \mathbf{p} \in S \quad (15)$$

Note that, since the values of  $\boldsymbol{\varphi}_S|_{D_S}^{-1}$  are known in all nodes of  $\mathcal{M}_S$ , the new mesh on the target surface can be defined as

$$\mathcal{M}_T := \tilde{\boldsymbol{\psi}}^0(\mathcal{M}_S)$$

### 3. DISCRETIZATION OF THE LINKING-SIDES

Linking-sides are always defined by four logical sides (each logical side can be composed by several edges). Therefore, they can be discretized using any standard structured quadrilateral meshing algorithm, for instance transfinite mapping (TFI) [13]. In order to apply the TFI method it is required that opposite logical sides have the same number of nodes.

It is important to ensure that high quality structured meshes are generated on the linking-sides. Note that these meshes determine the loops of nodes that are used later to generate the inner volume nodes disposed on several layers (see Figure 1(b)). Thus, if these surface meshes contain folded or low quality elements, then tangled, reverse oriented or low quality hexahedral elements will be obtained. This is of major importance for extrusion volumes defined by a curved sweep direction. For instance, consider the geometry presented in Figure 3. It is defined by a square cross-section which is swept along an  $\Omega$ -shaped path. In this case, the difficulty is to obtain a high quality structured mesh over the  $\Omega$ -shaped surface. If nodes are generated equidistant along the edges of the  $\Omega$ -shaped surface, then some segments of the structured surface mesh will cross over each other, see Figure 3(a). Thus, folded quadrilateral elements

## MESH PROJECTION BETWEEN PARAMETRIC SURFACES

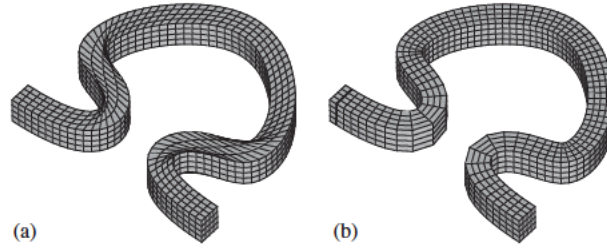


Figure 3. Discretization of a  $\Omega$ -shaped sweepable volume: (a) folded quadrilateral elements are obtained if equidistant nodes are placed along the edges; and (b) high quality elements are generated over the linking-sides if edge nodes are placed using the new edge-mesher.

will be obtained in the middle part of the surface mesh, and tangled hexahedral elements are generated inside of the sweep volume. To overcome this shortcoming, we have developed an edge-mesher procedure that is able to ‘follow’ nodes on the opposite edge of the surface. To this end, the distance between two opposite nodes across the surface is minimized. Using this procedure, consecutive joining segments will not cross each other on the surface, see Figure 3(b). It is important to note that no surface mesh smoothing is required with this procedure.

## 4. GENERATION OF INNER NODES AND ELEMENTS

Once all boundaries are meshed, inner nodes of the extrusion volume have to be generated. These nodes have to be placed on several layers along the sweep direction. Each layer is delimited by several loops of nodes that belong to the structured meshes of the linking-sides. In fact, for every layer there is one outer loop and one inner loop for each hole in the sweep volume. Note that the method developed in Section 2 cannot be used here because no surface parameterization,  $\phi_T|_{D_T}$ , is available for these layers. Hence, the method is extended to  $\mathbb{R}^3$  (see Reference [16] for a detailed analysis) and a weighted least-squares approximation is developed, similar to those proposed in References [3–5].

Assume that  $r - 1$  inner levels of nodes have been generated on the linking-sides along the sweep direction (see Figure 2(b)). Therefore,  $r - 1$  layers of inner nodes have to be generated. Let  $X_0 = \{\mathbf{x}_0^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$ ,  $X_r = \{\mathbf{x}_r^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$  and  $X_k = \{\mathbf{x}_k^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$  with  $k = 1, \dots, r - 1$  be the physical co-ordinates of the boundary nodes of the source surface (level 0), the target surface (level  $r$ ) and the  $k$ th level, respectively. For a given level  $k$ , with  $k = 1, \dots, r - 1$ , we look for a function  $\phi$  such that

$$\mathbf{x}_k^i = \phi(\mathbf{x}_0^i), \quad i = 1, \dots, m \quad (16)$$

As in Section 2,  $\phi$  is approximated by an affine mapping

$$\mathbf{x}_k = \phi(\mathbf{x}_0) \approx \mathbf{A}(\mathbf{x}_0 - \mathbf{x}_0^c) + \mathbf{x}_k^c \quad (17)$$

where  $\mathbf{x}_0$  and  $\mathbf{x}_k$  are points on the levels 0 and  $k$  respectively,  $\mathbf{x}_0^c = (1/m) \sum_{i=1}^m \mathbf{x}_0^i$ ,  $\mathbf{x}_k^c = (1/m) \sum_{i=1}^m \mathbf{x}_k^i$  and  $\mathbf{A}$  is a linear transformation with its origin at  $\mathbf{x}_0^c$ . In order to avoid a system of

normal equations with a singular matrix (see References [3, 16, 17] for details), the co-ordinate system proposed in Reference [3] is used

$$\bar{\mathbf{x}}_0 = \mathbf{x}_0 - (2\mathbf{x}_0^c - \mathbf{x}_k^c), \quad \bar{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x}_0^c \quad (18)$$

Then, (17) can be expressed as

$$\bar{\mathbf{x}}_k = \bar{\Phi}(\bar{\mathbf{x}}_0) \approx \mathbf{A}\bar{\mathbf{x}}_0 \quad (19)$$

where  $\bar{\Phi}$  is the expression of  $\Phi$  in the new co-ordinates (18). A least-squares fitting of the boundary data is performed in order to find a linear transformation, that minimizes

$$F(\mathbf{A}) = \sum_{i=1}^m \|\bar{\mathbf{x}}_k^i - \mathbf{A}\bar{\mathbf{x}}_0^i\|^2 \quad (20)$$

Since  $\mathbf{A}$  is a linear transformation, it can be written as a linear combination of a basis of the linear transformations from  $\mathbb{R}^3$  to  $\mathbb{R}^3$ :

$$\mathbf{A} = \sum_{i=1}^9 \lambda_i \mathbf{h}_i \quad (21)$$

where  $\lambda_i \in \mathbb{R}$ , with  $i = 1, \dots, 9$ , and given  $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ ,

$$\begin{aligned} \mathbf{h}_1(\mathbf{x}) &:= \begin{pmatrix} x_1 \\ 0 \\ 0 \end{pmatrix}, & \mathbf{h}_2(\mathbf{x}) &:= \begin{pmatrix} x_2 \\ 0 \\ 0 \end{pmatrix}, & \mathbf{h}_3(\mathbf{x}) &:= \begin{pmatrix} x_3 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{h}_4(\mathbf{x}) &:= \begin{pmatrix} 0 \\ x_1 \\ 0 \end{pmatrix}, & \mathbf{h}_5(\mathbf{x}) &:= \begin{pmatrix} 0 \\ x_2 \\ 0 \end{pmatrix}, & \mathbf{h}_6(\mathbf{x}) &:= \begin{pmatrix} 0 \\ x_3 \\ 0 \end{pmatrix} \\ \mathbf{h}_7(\mathbf{x}) &:= \begin{pmatrix} 0 \\ 0 \\ x_1 \end{pmatrix}, & \mathbf{h}_8(\mathbf{x}) &:= \begin{pmatrix} 0 \\ 0 \\ x_2 \end{pmatrix}, & \mathbf{h}_9(\mathbf{x}) &:= \begin{pmatrix} 0 \\ 0 \\ x_3 \end{pmatrix} \end{aligned} \quad (22)$$

The normal equations of the least-squares problem (20) defined by the basis (22) and the extension to  $\mathbb{R}^3$  of the scalar product (12) are

$$\mathbf{N}\boldsymbol{\lambda} = \mathbf{d} \quad (23)$$

where  $N_{i,j} = \langle \mathbf{h}_i, \mathbf{h}_j \rangle$  and  $d_i = \langle \mathbf{h}_i, \bar{\Phi} \rangle$ , with  $i = 1, \dots, 9$  and  $j = 1, \dots, 9$ .

Let  $\boldsymbol{\lambda}^0$  be the computed solution of the linear system (23). Using (18), (19), and (21), a least-squares approximation of the projection between the source surface and the  $k$ th level can be written as

$$\Phi_k^0(\mathbf{x}_0) := \mathbf{A}^0(\mathbf{x}_0 - 2\mathbf{x}_0^c + \mathbf{x}_k^c) + \mathbf{x}_0^c \quad (24)$$



## MESH PROJECTION BETWEEN PARAMETRIC SURFACES

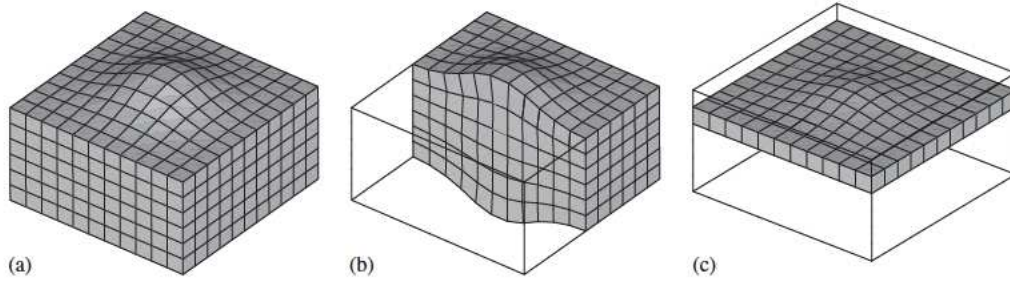


Figure 4. Extrusion volume with non-planar cap surfaces and inner nodes placed according to the weighted interpolation: (a) outer boundary mesh; (b) vertical cut showing the inner elements; and (c) a view of the fifth inner layer of elements.

where  $A^0$  is the linear transformation corresponding to the solution  $\lambda^0$  by (21), and  $\mathbf{x}_0$  is any point of the source surface mesh.

Note that a similar process could be defined using the target surface as initial surface instead of the source surface. Hence, the nodes of the  $k$ th level can be computed as

$$\phi_k^r(\mathbf{x}_r) := A^r(\mathbf{x}_r - 2\mathbf{x}_r^c + \mathbf{x}_k^c) + \mathbf{x}_r^c \quad (25)$$

where  $\mathbf{x}_r$  is any point of the target surface mesh.

Given two meshes  $\mathcal{M}_S$  and  $\mathcal{M}_T$  over the source and target surfaces, respectively, Equations (24) and (25) allow to obtain, according to Reference [5], the following weighted transformation:

$$\phi_k(\mathbf{p}) := \left(1 - \frac{k}{r}\right) \phi_k^0(\mathbf{p}) + \frac{k}{r} \phi_k^r(\tilde{\mathbf{v}}^0(\mathbf{p})) \quad (26)$$

where  $\mathbf{p} \in \mathcal{M}_S$ ,  $k = 1, \dots, r-1$ , and  $\tilde{\mathbf{v}}^0$  is the transformation defined in (15).

Finally, hexahedral elements are generated by joining the corresponding nodes between adjacent layers of quadrilateral meshes.

Figure 4 presents the discretization of an extrusion volume defined by two non-planar cap surfaces. It is composed by 726 elements. Figure 4(a) shows the surface mesh. Figures 4(b) and 4(c) show a detail of the shape of the inner hexahedral elements. Note that the weighted function (26) generates layers of elements with a smooth transition of the curvature from the source surface to the target surface. No smoothing was applied to obtain this mesh.

## 5. NUMERICAL EXAMPLES

In order to assess the quality of the sweep algorithm described above, six examples are presented. The geometry of the examples is defined using several commercial CAD packages. The user assigns the element size, and the application automatically determines the cap surfaces and the linear or non-linear sweeping direction. The first example shows the discretization of a mill head with linear sweeping axis and rotated cap surfaces, see Figure 5(a). It consists of 1170 hexahedral elements. As it can be seen, a unstructured quadrilateral mesh is generated over the cap surfaces. Note that, although the sweeping axis is twisted, high quality elements are generated without *a posteriori* mesh smoothing.

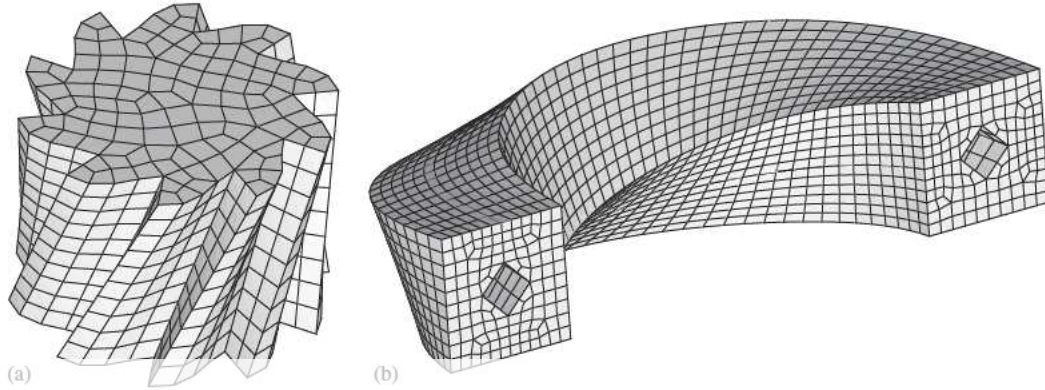


Figure 5. (a) Mill head with twisted sweep path; and (b) one-hole extrusion volume with twisted and curved sweep path.



Figure 6. Volume with varying elliptical cross-sections along a twisted sweep path: (a) final mesh; (b) layer of hexahedral elements at the fourth level; and (c) middle layer of hexahedral elements.

In the second example, a source surface defined by two squares centred at the same point with sides in a ratio of 0.4 and rotated  $45^\circ$  is extruded following a twisted and curved sweep path. The final mesh of 5104 hexahedral elements is depicted in Figure 5(b). As it can be seen, a non-structured quadrilateral mesh is generated over the cap surfaces. Note that, although the sweeping axis is twisted and curved, high quality elements are generated without *a posteriori* mesh smoothing.

The third example shows the discretization of an extrusion volume defined by varying and rotating cross-sections along the sweep path. These cross-sections are elliptical-shaped with different size, and just on the middle of the extrusion path becomes circular. Moreover, the cap surfaces are rotated  $90^\circ$ . The final mesh has 2373 elements, see Figure 6(a). It is important to point out that neither a surface smoothing nor a global smoothing are applied in this case. In order to show the quality of elements inside the volume, Figure 6(b) shows the fourth layer of hexahedral elements, and Figure 6(c) depicts the middle layer with one circular bounding loop.



## MESH PROJECTION BETWEEN PARAMETRIC SURFACES

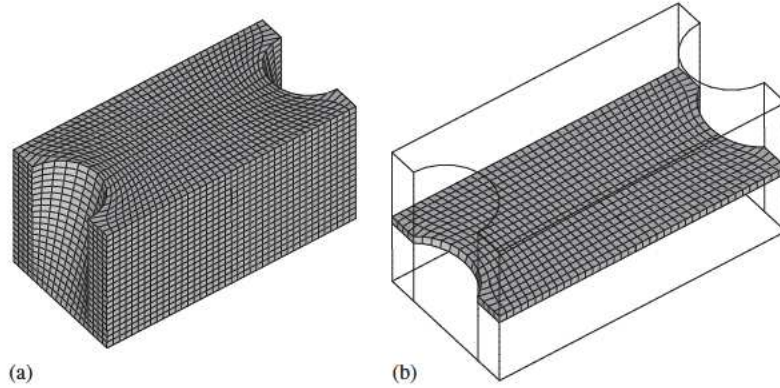


Figure 7. Sweep volume defined by a convex source surface and a target surface with two concavities: (a) obtained mesh; and (b) layer of hexahedral elements at the 10th level.

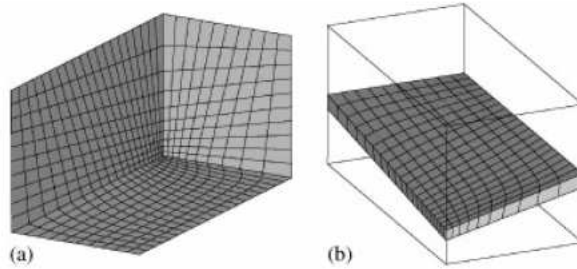


Figure 8. Discretization of a parallelepiped with non-uniform element size distribution: (a) detail of the final mesh at the corner where high element concentration is prescribed; and (b) view of the fourth inner layer of hexahedral elements.

The fourth example shows an application of the developed algorithm to an extrusion volume defined by two non-affine cap surfaces. In this case, a smoothing algorithm has been applied on the target surface mesh. Note that in this example the source surface is convex and the target surface has two concavities. Therefore, folded quadrilateral elements appear near the concavities of the target surface. In order to unfold these quadrilateral elements it is mandatory to smooth the target surface mesh. For this example we have used the smoothing technique presented in Reference [18]. Figure 7(a) shows the discretization of the geometry composed by 16 000 elements. Figure 7(b) shows the tenth inner layer of elements. Although the cap surfaces are not affine, no additional 3D global smoothing algorithm has been applied in this case.

In the fifth example, the discretization of a parallelepiped with a non-constant element size distribution is presented. A high element concentration is prescribed at one corner of the source surface. Hence, the boundary loops of nodes of the cap surfaces are not mutually affine. Figure 8(a) shows the final mesh composed by 2000 elements. Figure 8(b) shows a view of the fourth inner layer. Two smoothing steps were required: the first one to smooth the target surface mesh, and the second one to improve the overall quality of the hexahedral mesh.

The last example shows the application of the developed algorithm to an extrusion geometry composed by several sweep volumes. Figure 9(a) presents the geometry model

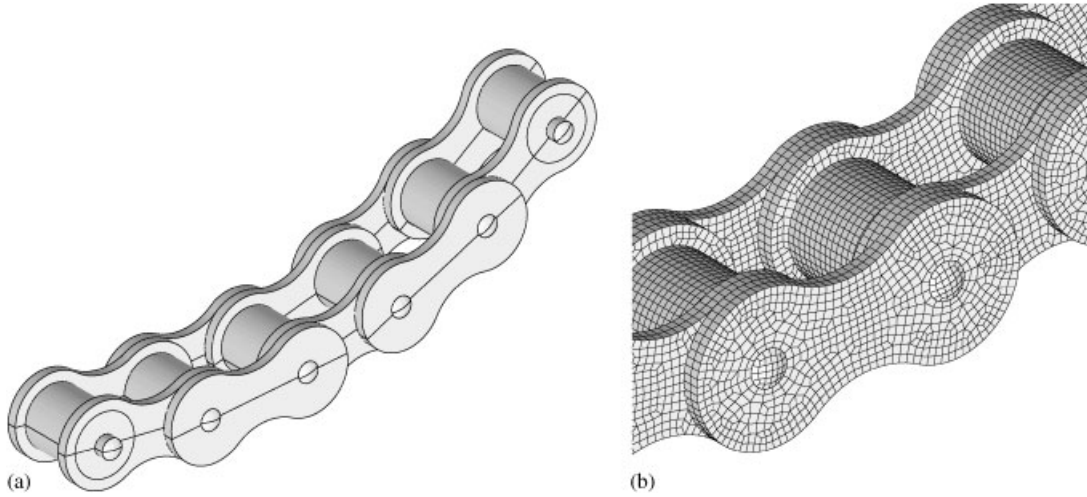


Figure 9. Power chain discretization: (a) block decomposition of the CAD model; and (b) hexahedral mesh.

of a power chain. The final mesh has 33 940 hexahedral elements. A detail of the obtained discretization is presented in Figure 9(b). Note that a conformal mesh is generated over the shared surfaces.

## 6. CONCLUSIONS

In this paper a new algorithm to project meshes between two topologically equivalent parametric surfaces has been presented. This projection is carried out by means of a least-squares approximation of an affine mapping defined between the parametric spaces of the surfaces. Once the new mesh is obtained in the parametric space, it is mapped to the target surface according to its parameterization. This projection algorithm has been extended to three-dimensional spaces, and it has been successfully implemented in a sweep tool able to mesh extrusion geometries defined by: (i) any CAD application; (ii) non-linear sweeping trajectories; (iii) non-constant cross section along the sweep axis; (iv) non-parallel cap surfaces; and (v) cap surfaces with different shape and curvature. The examples show that high quality hexahedral elements are generated, and that the layers of inner nodes are distributed in such a way that a smooth transition between the curvatures of cap surfaces is obtained. Moreover, they also illustrate that the developed algorithm, coupled with volume decomposition, can be successfully used to mesh a large class of three dimensional geometries.

Finally, it is important to point out that the global sweep algorithm could be generalized to multi-source/multi-target geometries following the cooper tool algorithm [5, 6]. Moreover, it may also be extended to multi-axis sweep directions according to Reference [19].

## REFERENCES

1. Tautges TJ. The generation of hexahedral meshes for assembly geometry: survey and progress. *International Journal for Numerical Methods in Engineering* 2001; **50**(12):2617–2642.
2. Thompson JF, Soni B, Weatherill N. *Handbook of Grid Generation*. CRC Press: Boca Raton, FL, 1999.



3. Knupp PM. Next-generation sweep tool: a method for generating all-hex meshes on two-and-one-half dimensional geometries. In *7th International Meshing Roundtable*, Dearborn, Michigan, U.S.A., 1998; 505–513.
4. Knupp PM. Applications of mesh smoothing: copy, morph, and sweep on unstructured quadrilateral meshes. *International Journal for Numerical Methods in Engineering* 1999; **45**(1):37–45.
5. Blacker T. The cooper tool. In *5th International Meshing Roundtable*, Pittsburgh, Pennsylvania, U.S.A., 1996; 13–30.
6. Mingwu L, Benzley SE. A multiple source and target sweeping method for generating all-hexahedral finite element meshes. In *5th International Meshing Roundtable*, Pittsburgh, Pennsylvania, U.S.A., 1996; 217–225.
7. Staten ML, Canann SA, Owen SJ. BMSweep: locating interior nodes during sweeping. *Engineering with Computers* 1999; **15**(3):212–218.
8. White DR, Tautges TJ. Automatic scheme selection for toolkit hex meshing. *International Journal for Numerical Methods in Engineering* 2000; **49**(1–2):127–144.
9. Blacker TD, Stephenson MB. Paving—a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1991; **32**(4):811–847.
10. Cass RJ, Benzley SE, Meyers RJ, Blacker TD. Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering* 1996; **39**(9):1475–1489.
11. Sarrate J, Huerta A. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 2000; **49**(10):1327–1350.
12. Sarrate J, Huerta A. Automatic mesh generation of nonstructured quadrilateral meshes over curved surfaces in  $\mathbb{R}^3$ . In *3th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, Barcelona, Spain, 2000.
13. Haber R, Shephard MS, Abel JF, Gallagher RH, Greenberg DP. A general two-dimensional, graphical finite-element preprocessor utilizing discrete transfinite mappings. *International Journal for Numerical Methods in Engineering* 1981; **17**(7):1015–1044.
14. Goodrich D. Generation of all-quadrilateral surface meshers by mesh morphing. *Master in Science Thesis*, Brigham Young University, 1997.
15. Hatcher A. *Algebraic Topology*. Cambridge University Press: Cambridge, 2002.
16. Roca X, Sarrate J, Huerta A. A new least-squares approximation of affine mappings for sweep algorithms. In *14th International Meshing Roundtable*, San Diego, California, U.S.A., 2005; 433–448.
17. Roca X, Sarrate J, Huerta A. Surface mesh projection for hexahedral mesh generation by sweeping. In *13th International Meshing Roundtable*, Williamsburg, Virginia, U.S.A., 2004; 169–179.
18. Sarrate J, Huerta A. A new approach to minimize the distortion of quadrilateral and hexahedral meshes. In *4th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, Jyväskylä, Finland, 2004.
19. Miyoshi K, Blacker T. Hexahedral mesh generation using multi-axis cooper algorithm. In *9th International Meshing Roundtable*, New Orleans, Louisiana, U.S.A., 2000; 89–97.